EXHIBIT E

[redacted]

From: Brian Jones (AUTHORING SVS)
Sent: Wednesday, May 21, 2003 6:12 PM
To: peter.vogel@[redacted] Joe Andreshak
Cc: Martin Sawicki
Subject: RE: WordML documents

I'm going through the doc and making comments. It is extremely busy right now though, and will most likely take me a while to finish.

-----Original Message-----
From: Peter Vogel [redacted]
Sent: Tuesday, May 20, 2003 2:22 AM
To: Joe Andreshak
Cc: Brian Jones (AUTHORING SVS); Martin Sawicki
Subject: RE: WordML documents

Joe:

Thanks for the kind words. After I sent the document off, the one subject that I'm sorry that I didn't address was VBA support. I think that another section on the tags for VBA, including the support for flags indicating the presence of VBA would be an excellent addition.

I suspect that the only real problem we will have is deciding where a primer stops...

-----Original Message-----
From: Joe Andreshak [redacted]
Sent: Tuesday, May 20, 2003 7:08 AM
To: peter [redacted]
Cc: Brian Jones (AUTHORING SVS); Martin Sawicki
Subject: RE: WordML documents

Peter, I've gone through the primer now. Coming together very nice! I have attached it here with my comments in the doc. The one thing I know we would like to add is information people (like antivirus vendors) need to determine whether or not executable code is in the file ... or if someone wants to auto-generate a file with such code ... that information (a recent addition) is listed below. Please ask Martin for any needed details. I noticed the current document didn't mention VBA, so through this discussion, I'm sure something can be incorporated.

Martin, Brian, we should talk once you've had a chance to go through it.
A few things for you to be thinking about:

We should make sure we have the final namespaces in the document. So far, these are referenced...

http://schemas.microsoft.com/office/word/2003/2/wordml
http://schemas.microsoft.com/office/word/2003/2/auxHint
urn:schemas-microsoft-com:vml
urn:schemas-microsoft-com:office:office
http://schemas.microsoft.com/aml/2001/core

Also, is there any important chunk of the WordML that should be touched in a primer?
And, we should have someone who knows this well to go through the document to validate the accuracy.
Lastly, Peter has a number of tables at the back listing properties and tags from the WordML schema. Clearly, as the schema documentation gets updated we'd like to suck that into this document. Can we work up some XML magic to do that? ;-)

Joe

========== Modifications for antivirus help ==========

The AV community was concerned that it would be difficult/non-performant for their AV software to scan WordML for VBA, OCX and OLE. So, we added output flags that indicate the presence of VBA, OCX and OLE in a doc and also enable the doc containing such code to be opened.

Below is the spec to make these attributes No by default (and not required if set to No):

<SPEC>
All of these attributes will be represented in Word's main wordml namespace on the <w:wordDocument> element.

ATTRIBUTE:

w:macrosPresent - attribute indicating presence of VBA code or toolbar customizations in the doc. Possible values:

VALUES:

"yes" -- means there is a docSuppData tag in the file, *before* the body element is encountered. If Word encounters the "body" element without finding docSuppData, then we will treat the document as corrupt.

"no", or any other value, OR ABSENT -- there is no docSuppData tag in the file. If a docSuppData element is encountered in the file, Word will treat the file as corrupt.

If the w:macrosPresent attibute is missing, then it is assumed to be set to "no".

(note that docSuppData is used for VB projects, toolbar customizations, and other things I can't think of (same purpose as editdata.mso in
html))

ATTRIBUTE:

w:embeddedObjPresent - attribute indicating the presence of one or more embedded OLE objects

VALUES:

"yes" -- means there is a docOleData tag in the file, *before* the body element is encountered. If word encounters the <body> element without finding docOleData, then we will treat the document as corrupt.

"no", or any other value, OR ABSENT -- means there is no docOleData tag in the file. If a docOleData element is encountered in the file, we will treat the file as corrupt.

If the w:embeddedObjPresent attribute is missing, then it is assumed to be set to "no".

ATTRIBUTE:

w:ocxPresent - (read carefully, this is sorta different) - indicates possible presence of OCX objects in the content.

VALUES:

"yes" -- there *may* be OCX objects present in the file. They will be located throughout the document content. The AV software must scan the entire document in order to locate their locations. Word will *never* write out "yes" if there are no OCX's in the file, but we will not treat the document as corrupt if this attribute is "yes" and we don't find any OCX objects

"no", or any other value, OR ABSENT -- means that there are *no* OCX objects in the file.

If any OCX is found in the file, the document is treated as corrupt. (this is the same as the other attributes)

If the w:ocxPresent attribute is missing, then it is assumed to be set to "no".

(The only real difference between this attribute and the others is that we won't strictly enforce the "yes" -- if they say "yes" there's an OCX but there isn't, we won't treat the document as corrupt for this attribute.

</SPEC>

-----Original Message-----
From: Peter Vogel [redacted]
Sent: Tuesday, May 13, 2003 9:05 AM
To: Joe Andreshak

Joe:

Here are the two WordML documents:
WordML Primer
WordML for Developers (same as above, with code, and a slightly different organization)

There are three things that I wasn't able to do in the document, none of which I think are fatal:

1. I would have like to include a comprehensive list somewhere of the content of the instrText tag/attribute and what fields are generated from each. I could generate this myself by working through the various field options, but I'm out of time.

2. I couldn't figure out how an alphabetized list gets the values A, B, C, etc. assigned to it. In other words, I couldn't spot the difference between a listDef for a numbered list with numbers and a numbered list with letters.

3. I have a set of tables in my WordML document that list off various elements with a brief description. Mostly, I've pulled those descriptions from the annotations in the WordML schema. However, there remain these 9 items that I couldn't figure out and I'm hoping you can provide descriptions for. I suspect that most of them just reveal my ignorance of Word.

Style tag child elements_____ autoRedefine
personal
personalCompose
personalReply

trPr tag child elements_____ cnfStyle

tcPr tag child elements_____ tcFitText

ListDef tag child elements_____ plt listStyleLink

lvl tag child elements_____
nfc
lvlText
suff
lvlPicBulletId

Peter Vogel
[redacted]